

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
14 December 2000 (14.12.2000)

PCT

(10) International Publication Number
WO 00/76122 A2

(51) International Patent Classification⁷: H04L 12/00

(21) International Application Number: PCT/US00/15251

(22) International Filing Date: 2 June 2000 (02.06.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/325,117 3 June 1999 (03.06.1999) US

(71) Applicant: FUJITSU NETWORK COMMUNICATIONS, INC. [US/US]; 2801 Telecom Parkway, Richardson, TX 75082 (US).

(72) Inventors: MCNEIL, Roy; Two Meadow Ridge Road, Warwick, NY 10990 (US). PIETSCH, Bruce; 14 Chapel Court, Orangeburg, NY 10962 (US).

(74) Agents: LEBOVICI, Victor, B. et al.; Weingarten, Schurgin, Gagnebin & Hayes LLP, Ten Post Office Square, Boston, MA 02109 (US).

(81) Designated States (*national*): AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:

— Without international search report and to be republished upon receipt of that report.

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

WO 00/76122 A2

(54) Title: LAN EMULATION USING PAIRED UNICAST AND BROADCAST VIRTUAL CONNECTIONS

(57) Abstract: An emulated local area network (ELAN) bridging technique employs unicast and multicast virtual connections (VCs) among ELAN bridges (72). "Known" ELAN frames, i.e., frames whose MAC addresses have been learned, are exchanged among the bridges on the unicast VCs. The multicast VCs are used to multicast "broadcast" and "unknown" ELAN frames, where "broadcast" frames contain a broadcast MAC address and "unknown" frames contain MAC addresses that have not been learned. Each bridge maintains an association (80) between each unicast VC and the multicast VC for the same bridge. For frames received from another bridge on a multicast VC, an association (82) is established between the source MAC address and the unicast VC associated with the multicast VC on which the frame was received. The MAC-VC association is used for forwarding received frames to other bridges in the ELAN using the specified unicast VCs.

TITLE OF THE INVENTION

**LAN Emulation Using Paired Unicast and Broadcast Virtual
Connections**

CROSS REFERENCE TO RELATED APPLICATIONS

5 --Not Applicable--

STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR
DEVELOPMENT

 --Not Applicable--

BACKGROUND OF THE INVENTION

10 The present invention relates generally to the field
of data networks, and in particular to techniques for
performing Local Area Network (LAN) bridging over a wide-
area network (WAN) such as an Asynchronous Transfer Mode
(ATM) network.

15 LAN emulation or LANE is a technique for providing
data communication services between devices residing on
different LAN segments that are interconnected by a long-
distance or wide-area network (WAN). LANE effectively
hides the underlying WAN network from the devices, and
20 thus enables the devices to communicate using only their
native LAN protocol. Several important benefits can be
achieved using LANE, notably a measure of backwards
compatibility. By supporting LANE, newer WAN networking
equipment can successfully interoperate with devices
25 having only LAN interfaces. Customer investments in LAN
equipment can be protected, easing customer acceptance of
newer networking technology.

One common environment in which LANE is used is in ATM-based networks. In ATM networks, data is transferred in the form of fixed-length cells along pre-established "Virtual Connections" or VCs. This operation is very different from the operation of most LANs. One widely used LAN protocol, for example, is the Ethernet protocol. Ethernet networks generally employ a broadcast transmission medium, such as a multi-drop coaxial cable or twisted conductor pairs. Data is sent in the form of frames, each of which includes a destination address identifying the network node to which the frame is being sent. All nodes are required to "listen" for transmissions that contain the address of the node as the destination address. The LANE mechanism in an ATM WAN is responsible for presenting appropriate interfaces to LAN segments connected by the WAN, and performing operations on behalf of the LAN devices using mechanisms available in the ATM network.

LANE bridging involves the forwarding of frames from one LAN segment to another across the WAN. Emulated LAN (ELAN) bridges are responsible for learning the locations of other ELAN bridges in the WAN, maintaining a set of mappings from LAN destination addresses to VCs by which the destination nodes can be reached, and using the mappings to forward frames through the WAN toward destination nodes. Another function in which ELAN bridges participate is the flooding or broadcasting of frames containing "unknown" destination addresses (i.e., addresses identifying nodes whose physical locations in the WAN are unknown) to all possible destination LAN segments. Once a destination address becomes known,

subsequent frames are unicast transmitted using the now-known physical address information.

In one standard LANE configuration, an ELAN includes a group of geographically separated LANs that communicate with each other over an ATM network. The ELAN is structured according to a client-server model. Each ELAN bridge contains a LAN emulation client (LEC) that communicates with various LAN servers to carry out LANE operation. Among the LAN servers are a LAN emulation server (LES), a Broadcast and Unknown Server (BUS), and a LAN Emulation Configuration Server (LECS).

The main function of a LEC is to forward data frames across the ELAN to a peer LEC on another LAN segment. As part of this function, the LEC must learn and maintain mappings between layer-2 addresses (such as MAC addresses) and ATM addresses.

The LES enables the LEC to join an ELAN and to request an address resolution service, commonly referred to as LAN emulation address resolution protocol or LE_ARP. The BUS provides data delivery service on behalf of the LECs for broadcast, multicast, and "unknown" frames (i.e., frames destined for nodes whose location in the network are not known to a service-requesting LEC). The LECS provides the LECs with configuration information such as the network address of the LES, an ELAN identifier value, and an allowed maximum frame size.

Standard LANE bridging as described above has several advantages, including the use of dedicated, specialized broadcast servers to perform the broadcast function, rather than burdening each ELAN bridge with the broadcast capability. Also, standard LANE bridging scales up well. As a network grows, additional LESs,

LECSs, and BUSs can be added as needed to support the additional demands. Unfortunately, however, standard LEC bridging does not scale down quite as well, because at least one of each server type must be included in even the smallest ELAN. If the ELAN is small enough, the server functionality may not be fully utilized, so that network cost effectiveness is not optimal. It would be desirable to support LAN emulation in smaller networks in a more cost-effective manner.

BRIEF SUMMARY OF THE INVENTION

In accordance with the present invention, a method of performing emulated local area network (ELAN) bridging is shown that does not require a dedicated LAN emulation server or broadcast and unknown server, and thus can be advantageously employed in smaller, cost-sensitive network configurations.

According to the disclosed bridging technique, unicast and multicast virtual connections (VCs) are maintained among ELAN bridges. Each unicast VC is used by a pair of bridges to exchange "known" ELAN frames, i.e., frames whose MAC addresses have been learned by the sending bridge. The multicast VCs are used by the bridges to multicast "broadcast" and "unknown" ELAN frames to the other ELAN bridges, where "broadcast" refers to frames sent on a broadcast MAC address and "unknown" refers to frames containing MAC addresses that have not been learned by the sending bridge.

Each ELAN bridge maintains an association between each unicast VC to another ELAN bridge and the multicast VC for the same ELAN bridge. When a bridge receives frames from another bridge on a multicast VC, it establishes an association between the source MAC address

and the unicast VC associated with the multicast VC on which the frame was received. This association may be conveniently stored in a lookup table indexed by MAC addresses. When a frame is received for forwarding to
5 another bridge, the lookup table is consulted to determine whether the frame is associated with a unicast VC. If an entry is found, then the frame is forwarded on the unicast VC indicated by the entry.

The disclosed technique achieves cost advantages
10 because it does not rely on separate LAN servers (e.g. LES, LECS and BUS) and their attendant fixed costs. Broadcasting is achieved through the use of multicast VCs, which are inherently supported by ATM devices, such as ATM switch fabrics, that are present in ATM networks.

15 Other aspects, features, and advantages of the present invention are disclosed in the detailed description that follows.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWING

Figure 1 is a block diagram of an ATM network access
20 device incorporating emulated LAN bridging functionality in accordance with the present invention;

Figure 2 is a block diagram of a LAN interworking card in the network access device of Figure 1 in which the emulated LAN bridging functionality is provided;

25 Figure 3 is a block diagram of a wide-area network including an emulated LAN in accordance with the present invention;

Figure 4 shows translation and lookup tables used by one emulated LAN bridge in the network of Figure 3 to
30 store learned MAC address information for forwarding frames in the network of Figure 3; and

Figure 5 shows translation and lookup tables used by another emulated LAN bridge in the network of Figure 3 to store learned MAC address information for forwarding frames in the network of Figure 3.

5

DETAILED DESCRIPTION OF THE INVENTION

Figure 1 shows a network device for enabling access to an Asynchronous Transfer Mode (ATM) network running over a Synchronous Optical Network (SONET) transport network. SONET operation is provided by a Synchronous Transfer Mode (STM) line unit 10 interfacing to fiber optic cables 12-1 and 12-2. The cables 12 connect the network device to other devices in the network, for example in separate point-to-point segments or in a ring topology. The STM line unit 10 converts data signals formatted as Synchronous Transport Signal-N (STS-N, where for example N equals 1, 3 or 12), appearing on service-side ports 14, to Optical Carrier-N (OC-N, where for example N equals 3, 12 or 48) on the cables 12.

The network device includes STM service units (STM SUs) 16 that provide STM interfaces to external devices that require access to the SONET network. The STM service units 16 interface directly with the STM unit 10 via corresponding ones of the service-side ports 14.

The network device also includes ATM service units 18 and Interworking service units 20, which interface to the STM line unit 10 via an ATM interface unit 22. The ATM interface unit 22 includes ATM switch fabric logic, and provides ATM transport for the ATM service units 18 and the Interworking service units 20, via the STM unit 10 and the SONET network. The ATM service units 18 provide ATM interfaces to external ATM devices that require access to the SONET network. The Interworking

service units 20 provide other types of interfaces to non-ATM devices for inter-network operations. One example of an interworking service unit is a Local Area Network (LAN) service unit, which provides Ethernet
5 interfaces to the SONET network. As described below, the LAN service unit provides Ethernet bridge functionality and LAN emulation capability.

Figure 2 shows a block diagram of a LAN service unit, which is one type of interworking service unit 20.
10 PHY/MAC circuitry 30 interfaces to four separate Ethernet transmission lines 32-1 through 32-4 via corresponding ports 33-1 through 33-4. 10BaseT or 100BaseT Ethernet frames are received by the PHY/MAC circuitry 30, and outgoing frames are transmitted in either a full or a
15 half-duplex fashion. The PHY/MAC circuitry 30 properly terminates the transmission media 32 while providing electrical isolation between the media 32 and the remainder of the circuitry on the LAN service unit. Within each port 33, PHY circuitry performs clock and
20 data recovery, tracks link status, and transfers received frame data to a MAC device also residing in the port 33. The MAC device checks frame validity, and identifies frames that contain errors. Partial frame data is stored in a 256 byte receive FIFO within the MAC device. The
25 MAC device also contains a transmit FIFO for transmit buffering. The receive and transmit FIFOs for each segment 32 interface to DMA logic 34 used to transfer frames to and from other components on the LAN service unit.

30 The DMA logic 34 services the ports 33 on a time division multiplexed access basis. The DMA logic 34 transfers frames between the MAC FIFOs and two packet-

processing units (PPUs) 36-1 and 36-2. Specifically, the DMA logic 34 transfers frames to and from packet memory 38 in each PPU 36. The DMA logic 34 contains an internal cross-connect matrix that allows for flexible assignment of Ethernet ports 33 to the PPUs 36. Each PPU 36 processes two of the four Ethernet ports 33.

The DMA logic 34 also transfers frames between the PPUs 36 and a system segmentation and reassembly (SAR) device 40, such as an ATOM4+™ device available from Toshiba, Inc. The DMA logic 34 also provides a communication path between the PPUs 36 and a CPU subsystem 42.

When the DMA logic 34 receives a MAC frame from a port 33, it creates a Buffer Descriptor and places it in packet memory 38 along with the received frame. The Buffer Descriptor contains information such as Ethernet source port, frame length, error status, frame data checksum, etc. The DMA logic 34 manipulates frame pointers on queues in order to "move" the frames from one component to another. The queues are stored in a queue memory 44. The queue memory contains the following queues for each of the four Ethernet ports 33:

1. Host Receive (RX) and Transmit (TX). Used to transfer frames between the PPUs 36 and the CPU subsystem 42.
2. Ethernet RX and TX. Used to transfer frames between the PHY/MAC circuitry 30 and the PPUs 36.
3. SAR RX and TX. Used to transfer frames between the PPUs 36 and the system SAR 40.
4. Free Buffer. Used to keep track of memory buffers that may be used to store frame data.

Each PPU 36 contains a Forwarding Engine (FE) 48, which services up to two Ethernet ports 33. Logically, each FE 48 behaves as two separate processing units.
5 Each processing unit within an FE 48 can function as either a Virtual Circuit (VC) based ELAN bridge or a LAN Emulation Client (LEC) attached bridge.

During receive frame processing, frame pointers are passed between the DMA logic 34 and the FEs 48. Each
10 pointer corresponds to a 128-byte page of packet memory 38. The DMA logic 34 places a frame pointer on the Ethernet RX queue after a frame is fully received by the DMA logic 34. The FE 48 examines the frame pointer, performs frame processing on the corresponding data in
15 packet memory 38, and then instructs the DMA logic 34 to move the frame pointer to the appropriate output queue, such as the SAR TX queue. The FE 48 receives only one pointer per frame to be processed. Additional pointers are stored in the DMA logic 34 for economy of pointer
20 movement; the information the FE 48 needs for processing is contained within the first page of the frame. Once the FE 48 instructs the DMA logic 34 where to place the pointer for a completely processed frame, the DMA logic 34 moves the remainder of the pointers onto the same
25 queue.

Receive frame processing in the FE 48 varies depending on the type of service, e.g. port mapped bridge, 802.1d bridge, or LEC attached bridge. Generally, frame processing commences with the reading of
30 the Buffer Descriptor and MAC header information. The Buffer Descriptor tells the FE which logical processing unit should service the incoming frame, and whether the

frame contains an error. During frame processing, the header portion of the frame is manipulated in packet memory 38, while the payload portion of the frame remains static.

5 Receive frame processing by a FE 48 is considered complete when the FE 48 updates the Buffer Descriptor and writes encapsulation data for the frame back into packet memory 38. The FE 48 updates the Buffer Descriptor by
10 populating a Connection ID (CID) field, setting a Frame Check Sequence (FCS) status bit (preserve or drop), and indicating an offset to the start of frame data from the beginning of a buffer. The encapsulation data is used to form a corresponding frame including the frame payload for transfer over an ATM circuit via the system SAR
15 device 40, where the ATM circuit to be used is indicated by the value of the CID field.

 The apparatus shown in Figure 2 is capable of implementing up to four logical bridges, two per FE 48. Each FE 48 has two associated search tables (STs) 50 and
20 a Layer2/Layer3 lookup table (LUT) 52. Each ST 50 is a content-addressable memory (CAM) searchable by MAC address. The entries in each ST 50 contain pointers to locations in the LUT 52 associated with the same FE 48. The entries in the LUT 52 in turn contain information
25 describing how frames whose MAC addresses match those of the corresponding ST entries should be processed. For layer 2 (i.e., bridging) processing, the LUT 52 contains the CID, encapsulation type, and other service specific data for the frame.

30 MAC addresses are retrieved from the packet memory 38 and searched for in the corresponding ST 50. If a pointer to an entry in the LUT 52 is present, it is used

-11-

to retrieve the CID and other information from the LUT 52. This information is used to create the encapsulation data written back into packet memory 38 for the frame. Once frame processing is complete, the frame is placed on the SAR TX Queue to be transferred to the system SAR 40.

There are several exceptions to the above processing scenarios. These exceptions are as follows:

1. Pointers for frames containing errors are returned to the DMA logic 34 by the FE 48. No frame processing is performed by the FE 48. The DMA logic 34 returns the frame pointers to the Free Buffer Queue.
2. The search table lookup indicates that the current frame should be filtered. The frame is discarded by the FE 48.
3. The search table lookup indicates that the frame is destined for the CPU subsystem 42, also referred to as the Host. Bridge Protocol Data Units (BPDUs) are one type of frame that are destined for the Host. In this case, frame data is placed on the Host RX queue rather than the SAR TX queue.
4. The search table lookup indicates a "no match" condition, i.e., the search table has no LUT pointer for the MAC address being looked up. The resulting action depends on the type of service at the port. For VC-based ELAN bridging, the LUT is consulted for a CID of a broadcast Virtual Circuit (VC). This CID is placed in the Buffer Descriptor, and the frame is transferred to the system SAR 40 to be sent on the broadcast VC. For standard LAN Emulation (LANE) bridging, the frame is transmitted to the system SAR 40 to be sent to a Broadcast and Unknown Server (BUS) in the emulated LAN, and additionally an address

resolution process is carried out to obtain a mapping between the MAC address and a VC. Subsequent frames containing the MAC address are forwarded onto the VC to which the MAC address is mapped.

5 Frames destined for the ATM/SONET network are placed on the SAR TX queue for transfer to the system SAR 40. There are four SAR TX queues, one for each Ethernet port 33 (or one per bridge instance). Frames from each SAR TX queue are time-division multiplexed into a single input
10 queue within the system SAR 40. The system SAR 40 segments the frames and stores them as groups of ATM cells on VC queues within a cell memory 54.

 In the illustrated embodiment, the cell memory 54 has 4 MB of storage. Each VC queue in the cell memory 54
15 has a programmable list size, so that the available buffer space can be flexibly assigned among the VCs. The sum total of list sizes for all VCs within a given Virtual Path (VP) can be larger than the total amount of memory space allocated to the VP, in order to provide
20 statistical buffer gain. Once a VC queue reaches its programmed limit within the system SAR 40, subsequent frames destined for that VC are dropped. In addition, once the entire space allocated to a given VP is full, subsequent frames are dropped.

25 SCBI logic 56 (where SCBI stands for SAR Coprocessor Backplane Interface) provides an interface between the LAN service unit and the ATM interface unit 22 of Figure 1. The SCBI logic 56 has one interface to the system SAR 40, and another interface to the CPU subsystem 42. In
30 the illustrated embodiment, these interfaces conform to the UTOPIA standard, which specifies a multi-bit interface that provides efficient transfer of ATM cell

data. The CPU subsystem 42 contains its own SAR 58 to facilitate the segmentation and reassembly of frames on multiple VCs required by software executing in the CPU subsystem 42. In a preferred embodiment, the CPU
5 subsystem 42 employs the MPC860SAR microprocessor manufactured by Motorola, Inc.

For Ethernet sourced traffic, the SCBI logic 56 receives cells from the system SAR 40 and transmits them on a high-speed serial transmission line 60 to the ATM
10 Interface Unit 22 of Figure 1. The SCBI logic 56 also receives cells from the CPU subsystem 42, via the CPU SAR 58, and transmits these cells on the transmission line 60 to the ATM Interface Unit 22.

Cell-based traffic is received from the ATM
15 interface unit 22 over a high-speed serial transmission line 62. The SCBI logic 56 extracts the VPI/VCI and PT (Payload Type) fields of the incoming cells, and uses these values as inputs to a table whose entries indicate the cell type. The action taken depends on the cell
20 type, as follows:

1. A user data cell is translated through a VC Translation Table and stored in a cell buffer 64 for forwarding to the system SAR 40.
2. A LAN emulation control frame (as opposed to an in-band
25 frame) is placed untranslated into a cell buffer 66 for forwarding to the CPU subsystem 42.
3. Management cells are placed untranslated into the cell buffer 66 for forwarding to the CPU subsystem 42.

The system SAR 40 performs AAL5 reassembly of frames
30 from the cells it receives, and checks the integrity of the reassembled frames. In particular, the system SAR 40 checks for and flags the following conditions: (1) frames

too large; (2) frames having lengths different from the AAL5 frame length field; and (3) frames having CRC errors. Reassembled frames are placed in frame lists at the frame interface of the system SAR 40. The system SAR 40 attaches a CID, an Encapsulation Type field, and a Bridge ID to the beginning of each frame on the list. These fields are set up within the system SAR 40 by operating software when a new VC is provisioned within the system. The frames and frame lists are stored in the cell memory 54.

The DMA logic 34 transfers frames out of the system SAR 40 in a time division multiplexed access manner. From each frame, the DMA logic 34 forms a Buffer Descriptor based on the CID, Encapsulation Type, Bridge ID, frame length, and the fact that the frame entered from the ATM side of the LAN service unit. The frame is placed on the SAR RX queue for the appropriate logical bridge.

The PPU 36 that receives the frame from the DMA logic 34 processes the frame in a similar manner as for frames received from the Ethernet side. The frame may be destined for an Ethernet port 33 or Host software executing in the CPU subsystem 42. Each outgoing frame encountering a "no match" condition is simply forwarded to the Ethernet port 33 associated with the bridge. Decapsulation processing for multiprotocol encapsulation per RFC 1483 and LANE bridging is performed. Processed frames are placed on either the appropriate Ethernet TX Queue or the Host RX Queue.

The DMA logic 34 forwards outgoing frames to the MAC controllers in the respective ports 33 within the PHY/MAC circuitry 30. Each MAC controller contains a 256-byte

transmit FIFO used to buffer outgoing frames. The DMA logic transfers frames into the transmit FIFO from the packet memory 38. Whenever data is available in a MAC transmit FIFO, the corresponding PHY transmits the data onto the Ethernet media 32.

Figure 3 shows a network in which VC-based bridging is performed over an emulated LAN (ELAN) on an ATM network 70. The ATM network 70 can be considered a wide-area network (WAN), in which multiple ELAN bridges 72-A, 72-B, 72-C and 72-D are interconnected by a variety of virtual connections (VCs). Physically, the ATM network 70 may consist of several network access devices of the type shown in Figure 1 and Figure 2 above, interconnected by fiber cables and employing the STM transport. Each ELAN bridge 72 in Figure 3 includes functionality implementing an IEEE 802.1 bridge port 73 at an interface to a physical LAN segment 32, and a virtual port shown as an RFC 1483 encapsulation function 74 interfacing to the ELAN. This functionality is implemented in the manner described above with reference to Figure 2. For simplicity, only one ELAN is shown in Figure 3. Given that the LAN Interworking card of Figure 2 is capable of supporting up to four ELAN bridges 72, in general there may be multiple separate ELANS in the ATM network 70.

Each ELAN bridge 72 in Figure 3 is connected to every other ELAN bridge 72 by three distinct VCs, as follows:

VC1 - Multicast to all other bridges

VC2 - Multicast from bridge X

VC3 - Unicast to/from bridge X

In each bridge 72, there are separate pairs (VC2, VC3) for every other bridge 72 in the group. Each unicast VC is used between two respective ELAN bridges 72 for transferring frames to each other when the sending
5 bridge knows that the destination is reachable via the other bridge. Such frames are referred to herein as "known" frames. As indicated, each ELAN bridge 72 also has a multicast VC, which is used for sending "broadcast" and "unknown" frames to all the other ELAN bridges 72.
10 "Broadcast" refers to frames sent using a broadcast MAC address, and "unknown" refers to frames that the receiving ELAN bridge 72 does not know how to forward in the ELAN.

A basic function performed in each ELAN bridge 72 is
15 "learning" MAC addresses, in order to efficiently forward frames in the ELAN. When a MAC address is learned, it transitions from "unknown" to "known". Referring to Figure 3, an example is presented to illustrate the basic learning process. A LAN node 76-A has MAC address A and resides on LAN segment 32-A attached to ELAN bridge 72-A,
20 and a LAN node 76-B has MAC address B and resides on LAN segment 32-B attached to ELAN bridge 72-B. When node 76-B sends a frame to node 76-A over the unicast VC VC3 between the ELAN bridges 72-A and 72-B, bridge 72-A
25 stores a pairing between MAC address B and a connection identifier (CID) associated with VC3. This pairing is stored in a forwarding lookup table. When the bridge 72-A receives a frame from LAN node 76-A destined for LAN node 76-B, it consults the forwarding table using MAC
30 address B as the search key. The entry pairing MAC address B with the CID for VC3 is found, and the frame is forwarded to the ELAN bridge 72-B via the unicast VC VC3.

These learning and forwarding processes are used by all the ELAN bridges 72.

It is desirable that the ELAN bridges 72 also learn MAC addresses from multicast traffic, but this is more complicated. It is possible for the bridges 72 to learn associations between MAC addresses and incoming multicast VCs on which frames are received. However, this information by itself is insufficient for frame forwarding. The multicast VCs are unidirectional into a receiving ELAN bridge 72, and therefore it is not possible to forward a frame to another ELAN bridge 72 using the other bridge's multicast VC.

Figure 4 and Figure 5 illustrate conceptually a technique by which an ELAN bridge 72 learns MAC-address-to-VC mappings from received multicast traffic such that information useful for frame forwarding is obtained. The technique employs a translation function, represented by translation tables 80, and a lookup function, represented by lookup tables 82. The lookup tables 82 map MAC addresses to VCs for forwarding frames, as described above. The translation tables 80 are populated when the various VCs used in the ELAN are provisioned. The entries in each translation table 80 reflect respective pairings between each unicast VC and a corresponding multicast VC over which frames from the same remote ELAN bridge 72 are received. The translation tables 80 map both the unicast VC and the multicast VC in a pair to the unicast VC for the pair. For example, translation table 80-A residing in ELAN bridge 72-A (Figure 4) maps unicast VC3 to VC3, and also maps multicast VC2 to unicast VC3. Similarly, translation table 80-B residing in ELAN bridge 72-B (Figure 5) maps unicast VC3 to VC3, and maps

multicast VC1 to unicast VC3. Although not shown in the Figures, it will be understood that the tables 80 contain similar entries for the other ELAN bridges 72-C and 72-D, and that similar tables 80 are maintained in the ELAN bridges 72-C and 72-D.

During operation, when a frame is received from the ELAN on a VC, the VC is first translated via the translation table 80. The translated VC and the source MAC address from the received frame are used to create a mapping entry in the lookup table 82. In the case shown in Figure 4, in which both VC2 and VC3 are to be associated with MAC address B, the entry (MAC B, VC3) is created when the first frame from node 76-B is received via either VC2 or VC3. Subsequently, when a frame destined for node 76-B is received by ELAN bridge 72-A from the LAN segment 32-A, the lookup table 82-A is consulted and VC3 is identified as the VC on which the frame should be forwarded. The frame is then forwarded to ELAN bridge 72-B on VC3. Operation of the other ELAN bridges 72 is analogous.

Figure 4 and Figure 5 illustrate the paired-VC bridging technique in a conceptual manner. The technique can be realized in different ways. According to one implementation, the translation function represented by the translation tables 80 is realized by configuring the system SAR device 40 of Figure 2 to associate both the unicast and multicast VCs of a pair with the same connection ID (CID) for frames received from the ATM network. This configuration is performed during initialization. During operation, the CID becomes associated with the unicast VC for outgoing frames, i.e., frames being sent to the ATM network. The first time a

frame having a given MAC address is received from the ATM network on either VC, the CID is placed in a forwarding lookup table in association with the received MAC address. When a frame is to be sent to the MAC address across the ATM network, the CID is retrieved from the forwarding lookup table, and the system SAR 40 forwards the frame using the unicast VC.

A technique for LAN Emulation bridging using paired unicast and multicast virtual connections has been shown. It will be apparent to those skilled in the art that other modifications to and variations of the above-described technique are possible without departing from the inventive concepts disclosed herein. Accordingly, the invention should be viewed as limited solely by the scope and spirit of the appended claims.

CLAIMS

What is claimed is:

1. A method of performing emulated local area network (ELAN) bridging among a plurality of ELAN bridges in an ELAN, comprising:
 - maintaining the following virtual connections (VCs) among the ELAN bridges: (i) a unicast VC between each pair of ELAN bridges via which known ELAN frames are exchanged, and (ii) a multicast VC for each ELAN bridge via which broadcast and unknown ELAN frames are multicast to the other ELAN bridges;
 - maintaining, in each ELAN bridge for every other one of the ELAN bridges, an association between the unicast VC and the multicast VC for the other ELAN bridge;
 - establishing, in each ELAN bridge for source addresses obtained from frames received on each multicast VC, an association between each source address and the unicast VC associated with the multicast VC on which the frame was received; and
 - in each ELAN bridge for each received frame to be forwarded to another ELAN bridge, (i) determining a unicast VC on which the frame is to be forwarded by comparing a destination address contained in the received frame with the source addresses for which associations with unicast VCs have been established, and (ii) forwarding the received frame to the other ELAN bridge via the determined unicast VC.

2. A method according to claim 1, wherein:

maintaining associations between unicast VCs and

corresponding multicast VCs comprises maintaining a translation table, the translation table containing entries each operative to translate a multicast VC to the associated unicast VC;

5 establishing associations between source addresses and unicast VCs comprises creating entries in a lookup table, each entry being created upon receipt of a frame on a multicast VC, each entry including the MAC address from the received frame and a unicast address retrieved

10 from an entry in the translation table for the multicast VC on which the frame is received; and

 determining unicast VCs for forwarding frames comprises retrieving entries from the lookup table using the destination addresses from the received frames.

15

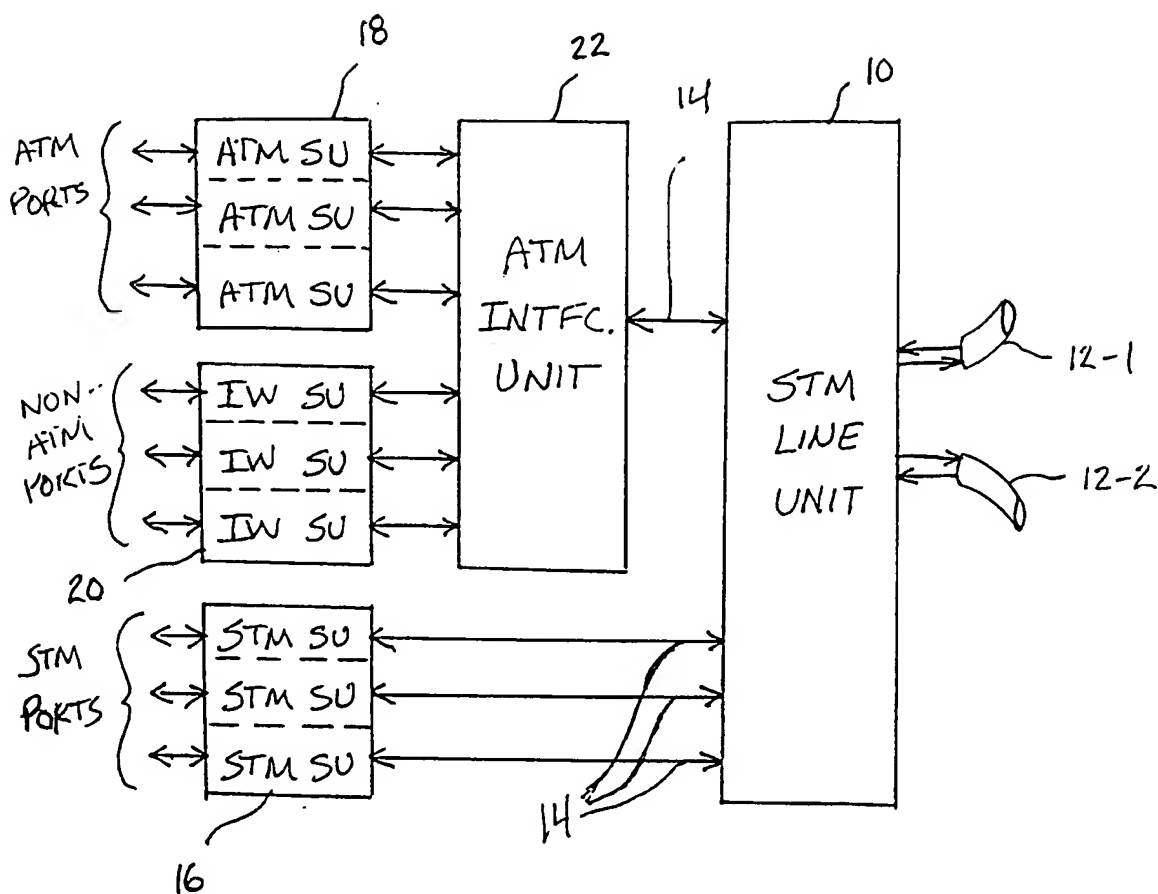
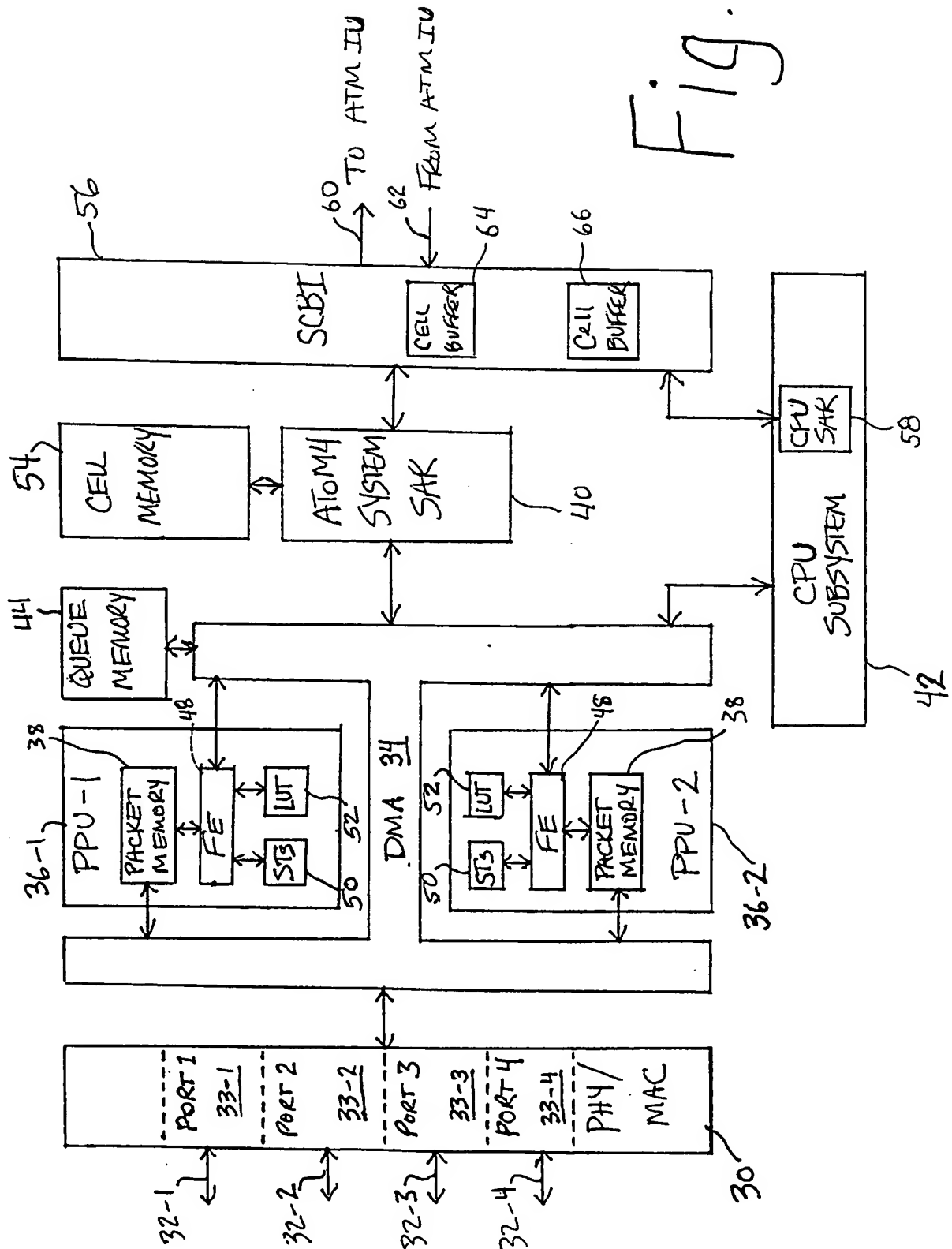


Fig. 1

Fig. 2



3/3

